# An Efficient Algorithm for Self-Organized Terminal Arrival in Urban Air Mobility

Joshua R. Bertram Peng Wei
*Iowa State University*
*Ames, IA 50011*
{bertram1, pwei}@iastate.edu

**Urban Air Mobility (UAM) is a concept for future air transportation where air taxis move passengers between vertical take-off and landing sites known as vertiports. While some form of a structured airspace is likely, it is expected that UAM systems will be required to deal with high traffic densities and will need to respond to conflicts due to a dynamically changing environment. Due to the difficulty of predicting demand of air taxi usage, it may also be more difficult to predict the required demand on any given vertiport. We explore an airspace design which can regulate the flow of aircraft landing at a vertiport, maintaining the aircraft in sequence until capacity is available at the vertiport. We demonstrate an implementation of the airspace design using a highly-efficient Markov Decision Process (MDP) based algorithm to provide separation and collision avoidance for a UAM terminal arrival sequencing problem. The vertiport maintains basic information about capacity and sequencing, and the aircraft seamlessly perform guidance in a self-organized distributed manner performing conflict avoidance while waiting to land.**

## I. Introduction

Urban Air Mobility (UAM) is a concept for future air transportation in which partially or fully autonomous air vehicles transport cargo and passengers through dense urban environments. To minimize the footprint required on the ground in urban environments, UAM concepts explore Vertical Take Off and Landing (VTOL) aircraft at sites known as vertiports. UAM aircraft are envisioned as navigating from vertiport to vertiport without established schedules or routes which will make terminal area capacity planning more difficult than for airports where scheduled commercial traffic and registered flight plans are available. In addition to UAM aircraft's inherent irregularity stemming from ad-hoc passenger requests, UAM aircraft can be expected to regularly change their flight path or destination due to changing passenger needs. For example, a passenger may wish to change the destination vertiport mid-flight due to a last-minute meeting cancellation. Similarly, a UAM vehicle may need to switch vertiports mid-flight due to a passenger pickup cancellation leading to the air taxi being dispatched to an alternate pickup location.

While we expect some form of terminal area air traffic control to manage each vertiport, we should also expect that the aircraft themselves respond to congestion appropriately by avoiding collision and adapting to circumstances as they develop. The system should also respond well in the case of communications datalink loss, which on aircraft can easily occur as the aircraft banks and turns temporarily blocking the propagation of radio signals due to the orientation of the airframe, airborne and ground based antennas.

In dealing with congestion, we also expect to transfer some ideas from commercial aviation airspace design that help organize incoming traffic into manageable flows that are designed to reduce conflict. One such idea is the use of metering fixes or arrival gates which serve as staging points that concentrate arriving aircraft into flows which can be sequenced. Aircraft are sequenced in order to provide an orderly flow of aircraft from the arrival gates to the landing areas. Whatever solution is implemented, it must also be simple and efficient enough that it can be deployed on embedded processors found on airborne vehicles.

In this paper we combine an airspace design with a real-time computational guidance algorithm that also performs collision avoidance. We propose a simple concentric rings based structure for the terminal airspace that regulates the flow of incoming aircraft. We demonstrate the concept using a 3D simulation environment which models the aircraft with a pseudo-6dof model and tracks Near Mid-Air Collisions (NMACs) during execution.

## II. Related Work

NASA, Uber and Airbus have been exploring the use of vertical takeoff and landing (VTOL) aircraft for Urban Air Mobility (UAM) [1–5]. In general, the UAM concept calls for UAM aircraft taking off and departing from small-scale airports known as vertiports where VTOL aircraft depart and arrive.

Researchers have examined structured airspace approaches. In [6] a vertiport is defined with two arrival and two departure metering fixes to separate climbing and descending traffic, though the focus of the paper is primarily on scheduling time of arrival for electric VTOL with limited battery charge rather than guidance or collision avoidance. Similarly, [7] also examines the time of arrival scheduling with a simple vertiport model. In [8], multiple vertiports over an urban area are modelled for purposes of scheduling network traffic, but the airspace structure itself is not. [9] discusses the overall UAM concept and describes example urban areas showing potential vertiport placement in example urban areas. In [10], an airspace concept known as streams is described which attempts to break airspace into separately managed, related streams of traffic bound for the same destination. In [11], the FAA performed a study of the North Texas Metroplex for the purposes of optimizing the airspace for precision based navigation. The study analyzes the airspace in the metroplex and describes a number of proposed changes to the airspace, including alterations to the standard terminal arrival route (STAR) metering fixes around DFW and DAL airports to better utilize the airspace and to eliminate identified inefficiencies. This report also examines airspace flows and describes trade-offs against different alternatives.

The method in this paper phrases the problem as a Markov Decision Process (MDP). MDPs are known to be difficult or intractable for large problems, but a recently discovered method for efficiently computing them was described in [12]. We use this underlying algorithm to provide guidance and collision avoidance while adding to it an airspace design to demonstrate efficient terminal arrival.

## III. Method

Figure 2 shows a conceptual design for a vertiport terminal area controller (VTAC). The terminal area airspace is composed of one or more concentric rings $r_i \in \{r_1, \cdots, r_n\}$. Each ring can support a limited number of aircraft known as the capacity $c_i$ which is determined by the circumference and ideal separation distance. A region around the vertiport within the innermost rings is reserved for vertical take off and landing (VTOL) operations at the vertiport and is assumed to be handled by some other controller outside the scope of this paper. The vertiport capacity $c_v$ is the number of aircraft the vertiport can simultaneously allow to land. The VTAC can thus support a total capacity of $C = c_v + \sum_{i=1}^{n} c_i$ aircraft.

An approach threshold is defined at a fixed radius from the vertiport beyond which approaching aircraft operate in free flight where they navigate to their goals while avoiding collision with other aircraft. $k$ approach gates are defined along the circumference of the approach threshold. If an aircraft wishes to land at the vertiport, it is assumed it communicates its intent to the vertiport, the vertiport verifies it has capacity, and the vertiport grants permission to enter the pattern. The approaching aircraft then proceeds to the nearest approach gate. Once the aircraft crosses the approach threshold, it is then under the control of the VTAC and is assigned a sequence number in a first-come-first-served manner.

The VTAC examines the capacity of the vertiport and the rings to determine where the aircraft should be assigned. In general, the aircraft will be assigned to the inner-most ring with spare capacity. If all rings are empty, then the aircraft will be directed to proceed to the vertiport for final approach.

All motion around the rings is in the same direction (assumed to be counter-clockwise in this paper). When an aircraft is assigned to a ring, the aircraft is responsible for entering the ring while maintaining separation from other aircraft. Other aircraft already in the ring are also responsible for maintaining separation. It is assumed a mechanism or sensor exists which can communicate or broadcast the position of all aircraft, possibly based off GPS positions, ground based radar, or ADS-B. The VTAC maintains a first-in-first-out FIFO queue for each ring so that the location of each aircraft within the airspace structure is known. When extracting the next entry from the FIFO queue, it will always be the aircraft with the lowest sequence number.

As aircraft flow through this airspace structure, first the inner most ring is filled, then the next ring is filled, and so on until all rings are full. If all rings are full, then the VTAC cannot allow any additional aircraft to enter the pattern and any new requests will be denied.

As the vertiport is able to accept new aircraft, aircraft in the inner most ring are selected for landing and are placed in a final approach queue. The aircraft then leave the inner most ring and approach the vertiport's VTOL region. Once

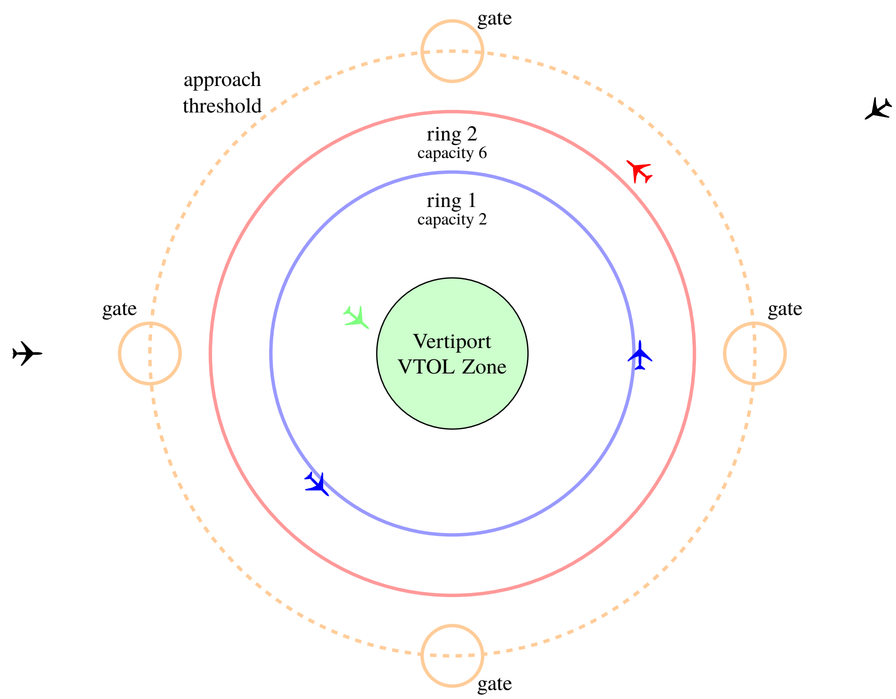


**Fig. 1 Terminal arrival airspace design showing two rings and four gates. Ring 1 has a capacity of 2 while ring 2 has a capacity of 6, meaning that the vertiport can keep up to 8 aircraft in the pattern while waiting for the vertiport to allow another aircraft to land. Traffic enters the airspace and approaches the closest arrival gate using free flight. Once each aircraft passes the approach threshold, it is assigned a sequence number and is under the direct control of the vertiport controller. The vertiport controller assigns the aircraft to a ring with spare capacity. The aircraft joins the ring avoiding traffic to ensure separation. As the vertiport has capacity to absorb additional aircraft, the aircraft with the next sequence number is directed to land at the vertiport. As inner rings gain spare capacity, aircraft are cycled from outer rings to inner rings in sequenced order. A separate vertiport control scheme (outside the scope of this paper) manages VTOL ascent and descent.**

they cross the threshold into this region, they are no longer under control of the VTAC and control transitions to the VTOL controller (outside the scope of this paper.)

The altitude of the rings could take different forms. We suggest that the outer rings be at the same or slightly higher altitude than inner rings. We envision that departing aircraft will depart below the altitude of the rings to ensure separation between arriving and departing aircraft. Alternatively, the departing VTOL aircraft could ascend above the altitude of the rings before moving into forward mode.

Now that we have defined the concept of operation of this vertiport terminal area controller (VTAC), we next describe an implementation of the controller based off a highly efficient algorithm that can perform collision avoidance while navigating to goals.

### A. FastMDP Algorithm

To implement a demonstration of this airspace design, we build on a recently proposed algorithm [12, 13] that can navigate through complex airspace environments to goals while avoiding collisions with aircraft and terrain. The problem is formulated as a Markov Decision Process (MDP) which accepts a set of positive and negative rewards. We provide a very brief overview of Markov Decision Processes and refer the reader to [13] for a more detailed description of the algorithm and aircraft dynamics. We can summarize by saying that a pseudo-6DOF model is used for the aircraft dynamics and that it is constrained with limits on the dynamics and actions to represent air taxi flight characteristics. While this model is not aerodynamically comprehensive, it is sufficient to describe aircraft motion suitable for examining our airspace design without loss of generality.

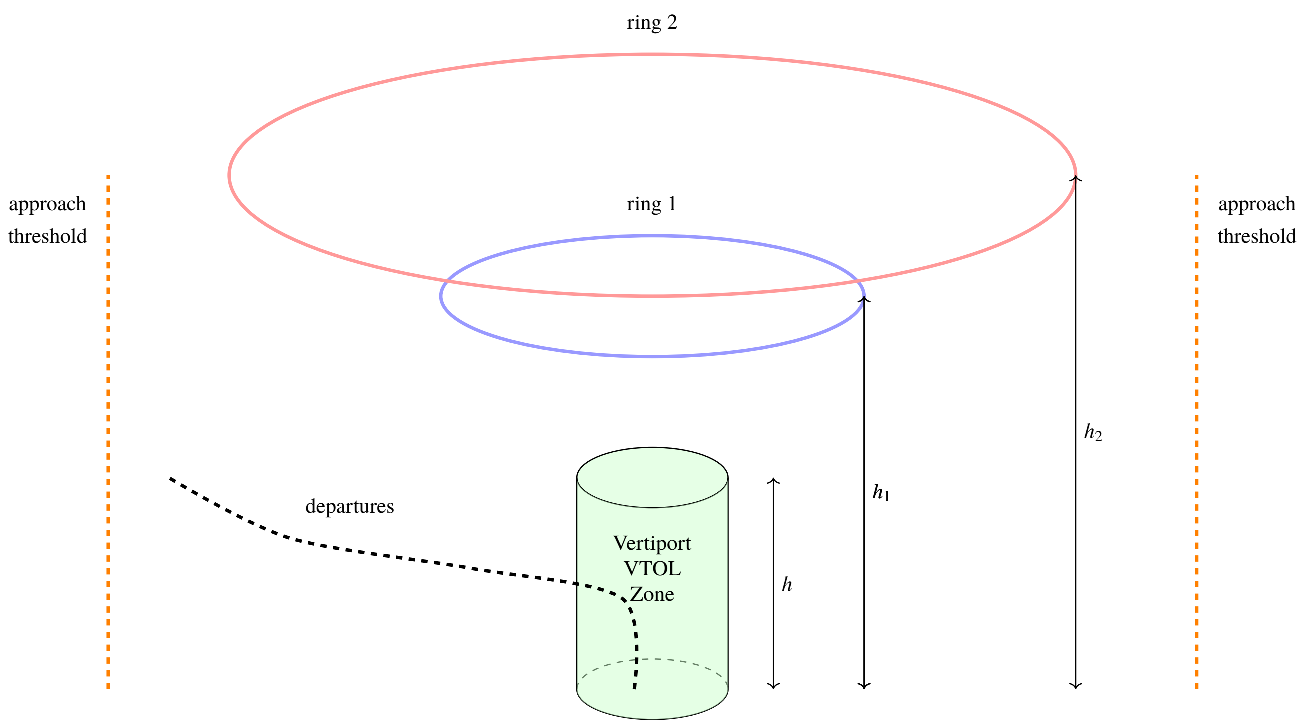


**Fig. 2 Profile view of vertiport terminal airspace showing relative altitudes of arrival and departures. The rings may be placed co-altitude (that is, $h_1$ and $h_2$ are the same) or outer rings may be at higher altitudes than inner rings. In this diagram, ring altitudes and diameters are not to scale. We envision that departing aircraft would use lower altitudes than ring traffic, but if required departing vehicles could ascend through the center of the rings to a higher altitude than the rings.**

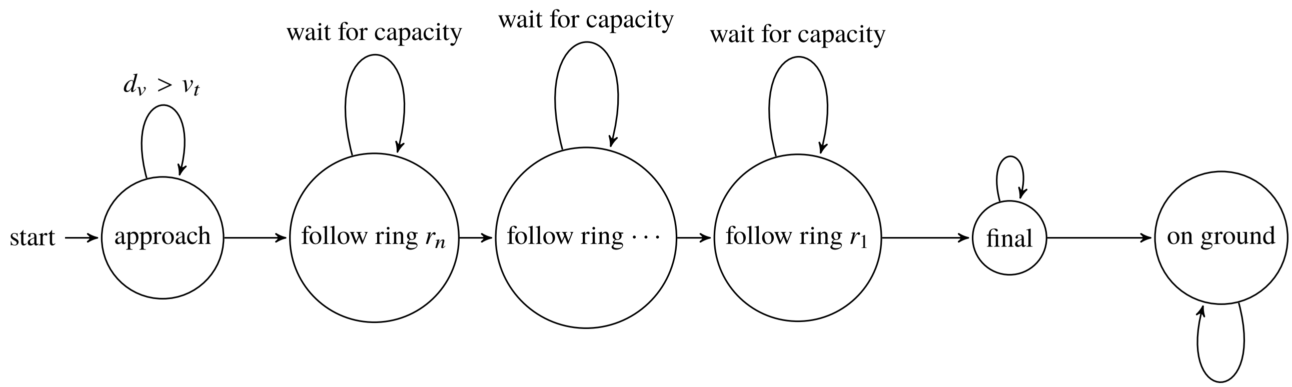


**Fig. 3 Per-aircraft arrival state machine. In the figure $d_v$ is the distance to the vertiport and $v_t$ is the approach threshold radius defined around the vertiport. The rewards defined for the MDP will differ depending on the per-aircraft state in this state machine.**

### B. MDP Formulation

Markov Decision Processes (MDPs) are a framework for sequential decision making with broad applications to finance, robotics, operations research and many other domains [14]. MDPs are formulated as the tuple $(s_t, a_t, r_t, t)$ where $s_t \in S$ is the state at a given time $t$, $a_t \in A$ is the action taken by the agent at time $t$ as a result of the decision process, $r_t$ is the reward received by the agent as a result of taking the action $a_t$ from $s_t$ and arriving at $s_{t+1}$, and $T(s_t, a, s_{t+1})$ is a transition function that describes the dynamics of the environment and capture the probability $p(s_{t+1}|s_t, a_t)$ of transitioning to a state $s_{t+1}$ given the action $a_t$ taken from state $s_t$. A policy $\pi$ can be defined that maps each state $s \in S$ to an action $a \in A$. From a given policy $\pi \in \Pi$ a value function $V^\pi(s)$ can be computed that computes the expected return that will be obtained within the environment by following the policy $\pi$.

The solution of an MDP is termed the optimal policy $\pi^*$, which defines the optimal action $a^* \in A$ that can be taken from each state $s \in S$ to maximize the expected return. From this optimal policy $\pi^*$ the optimal value function $V^*(s)$ can be computed which describes the maximum expected value that can be obtained from each state $s \in S$. And from the optimal value function $V^*(s)$, the optimal policy $\pi^*$ can also easily be recovered.

The same state space and action space as defined in [13] are used here. The reward function however is very different.

### C. Reward Function

The aircraft receives a different reward function depending on whether it is on approach, following a ring, or on final approach. In general, each aircraft receives a unique reward function. The reward function is designed to attract the aircraft to its goal(s) but treats all other aircraft and obstacles as risks that should be avoided.

We model this as a simple state machine for each aircraft as shown in Figure 3. Each state in the state machine will result in a different reward function (discussed below). First we define a set of primitives that will help describe the rewards that are define for each state in the state machine.

#### *1. Reward Primitives*

We model the goal(s) as a positive reward of 100. We allow the goal number and locations to vary over time, and each goal is modeled as a separate positive reward. We use a discount factor of 0.999 for the MDP. This provides a strong attraction to the goal globally over the state space.

To model the risk of colliding with another aircraft, we define a “risk well" as a negative reward of $-1000$ which decays at a rate of 0.995 for up to 2000 meters from the center of the well. For every aircraft, we place a risk well at the location the aircraft will be in 5 seconds, where it currently as at, and where it was 5 seconds in the past. This assumes that the other aircraft will maintain a constant heading and velocity and is used as a way to model the risk over the next 5 seconds.

When an aircraft is assigned to a ring, to discourage it from crossing the center region of the ring we define “terminal

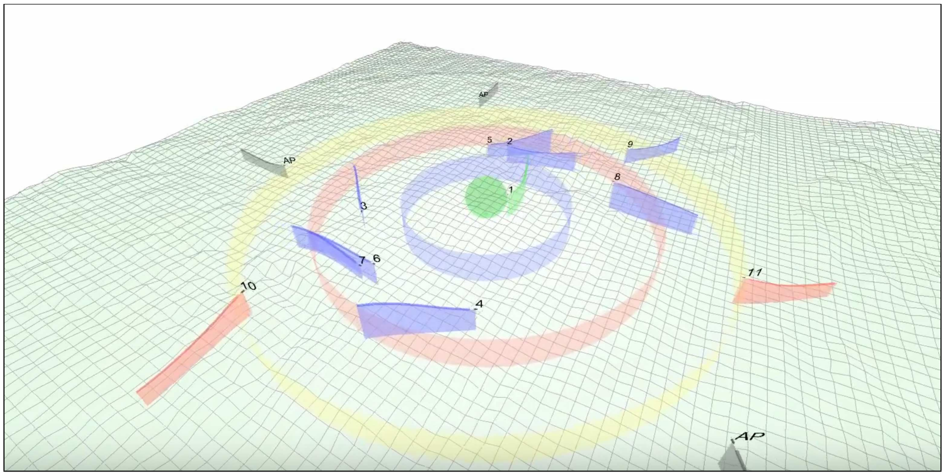


**Fig. 4 Screenshot of simulation showing a vertiport terminal area with two rings. Aircraft without numbers assigned are in the approach state; aircraft with numbers have been sequenced. Aircraft are color coded to match the ring that they are assigned to. Gray aircraft have not yet crossed the approach threshold colored yellow. Green aircraft are on final approach to the vertiport.**

well” as a negative reward of $-1000$ at the center of the ring with a radius of the ring’s radius. Thus if the aircraft strays inside the radius of the ring, it will receive a negative penalty and will desire to stay outside of the ring’s radius normally. Note that in the event that another aircraft is too close, the collision avoidance penalty will override the penalty of straying inside the ring. The ring’s radius is considered a soft constraint and collision avoidance is considered a hard constraint.

When an aircraft is assigned to a ring, we define a “future bearing” reward as a positive reward that is placed at a bearing that the aircraft will reach in the future if it were to follow the ring’s direction (e.g., counter clockwise.) For example, if an aircraft is currently at a bearing of 180 degrees (due south) of the vertiport, its future bearing will be some number of degrees $\alpha$ around the ring (perhaps at 140 degrees, south/south-east). As the aircraft moves around the circle changing its bearing with respect to the vertiport, the future bearing is also moving around the circle at the same rate. We model this as a positive reward of 100 placed on the circumference of the ring at the future bearing. This results in very smooth motion and is a simple and natural way to express the desired behavior for our aircraft. Note that the bearing and future bearing can be determined by the aircraft during flight and are not generated by the VTAC.

Next we describe how these reward primitives are used in each state to build the desired behavior in the airspace.

### *2. Rewards per state in state machine*

Depending on the state that the aircraft is in in Figure 3, the rewards that are provided to the MDP differ as follows:

1) When in the “approach" state:
   - A terminal well is placed at the vertiport so that the aircraft will avoid the airspace around the vertiport.
   - A positive reward is placed at each approach gate. The MDP solution will naturally draw the aircraft to the nearest goal.
   - Every other aircraft is modeled with risk wells to avoid collision with other aircraft.
2) When in the “ring following" state:
   - A terminal well is placed at the center of the ring with the same radius as the ring so that the aircraft will be discouraged from going within the ring’s radius.
   - A positive reward is placed at the future bearing of the aircraft. The MDP solution will naturally draw the aircraft to the future bearing location.
   - Every other aircraft is modeled with risk wells to avoid collision with other aircraft.
3) When in the “final" state:
   - A goal is placed at the vertiport so that the aircraft will be attracted to the vertiport VTOL zone.
   - Every other aircraft is modeled with risk wells to avoid collision with other aircraft.
4) When the aircraft reaches the VTOL zone it is removed from the simulation and is considered in the “on ground”

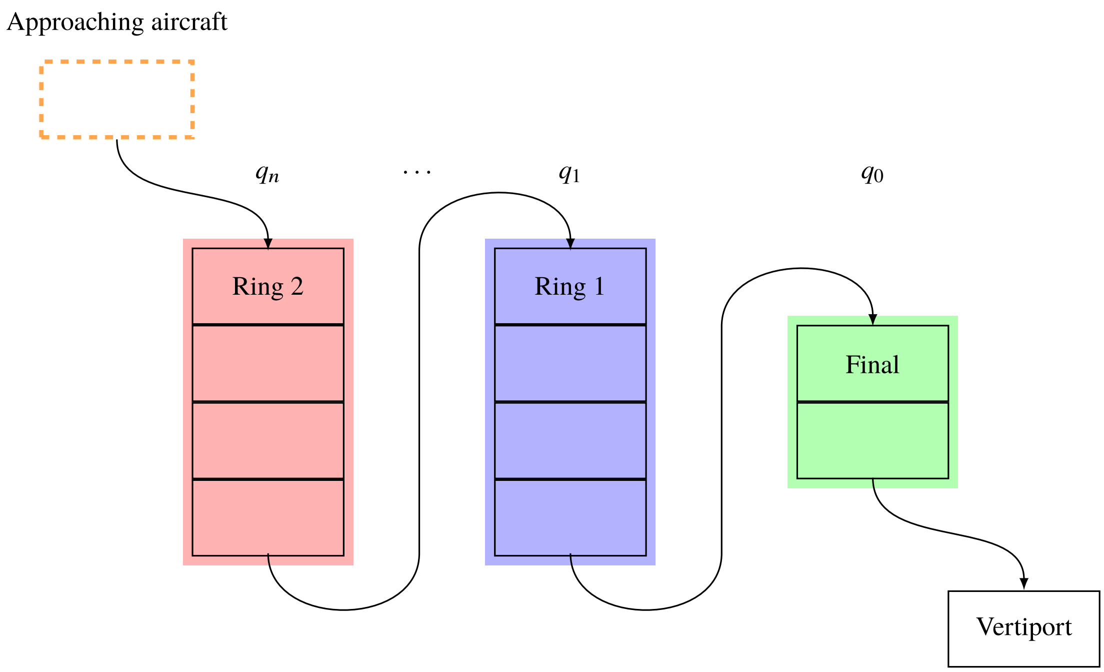


**Fig. 5 Depiction of aircraft flow through VTAC queues. Colors are chosen to be consistent with the ring colors from other diagrams.**

state. Any time taken to descend vertically to the airport is not considered in this simulation. In reality, some additional delay will be incurred by any ground service required to offload passengers, refuel/recharge, and load new passengers.

### D. Sequencing

As an aircraft transitions from the approach state to the final state, it is sequenced. Aircraft are sequenced in order that they cross the approach threshold. To implement sequencing, once an aircraft is sequenced we maintain a series of queues of all aircraft under VTAC control. When capacity is available at the vertiport, the aircraft with lowest sequence number is selected first.

While this method should be able to support other sequencing and prioritization schemes such as prioritizing landing for aircraft with the least amount of fuel/charge, this could potentially cause issues with aircraft moving through multiple rings and is left for future study.

### E. Ring Management

The VTAC maintains a queue for each ring $\{q_1, \cdots, q_n\}$ and a queue for the final approach to the vertiport $q_0$, will all queues referred to as $Q = \{q_0, q_1, \cdots, q_n\}$. Each queue is a first-in-first-out (FIFO) queue and has a fixed capacity. As aircraft cross the approach threshold, they are added to the outermost queue. At each update cycle, the vertiport attempts to move aircraft from outer rings to inner rings and from the innermost ring into the final queue. The queues are arranged as shown in Figure 5.

### F. Algorithm

The algorithm for the simulation for managing the vertiport is shown in Algorithm 1. We demonstrate this planner in a 3D aircraft simulation showing a perspective view of the aircraft, the rings, the approach threshold, and the vertiport as

**Algorithm 1** Vertiport simulation

```
procedure VertiportSim
    S_t ← randomized initial aircraft states
    A ← aircraft actions (precomputed)
    L ← aircraft limits (precomputed)
    S_{t+1} ← allocated space
    while aircraft remain do
        // Update the vertiport queues
        for q_i ∈ Q, ∀i = {0, ··· , n} do
            while not q_i.full do
                // Look for aircraft in outer rings to pull inward
                for q_j ∈ Q, ∀j = {i + 1, ··· , n} do
                    if not q_j.empty then
                        q_t ← q_j.remove()
                        q_i.add(q_t)
                        break out of for q_j loop
        // Aircraft states may have changed due to queue processing above
        for each aircraft do
            s_t ← S_t[aircraft]
            // Build peaks based off the flight phase
            if approach phase then
                P+ ← build pos rewards from arrival gates
                P− ← build neg rewards in Standard Positive Form to avoid collisions
            else if ring following phase then
                P+ ← build pos rewards from future bearing
                P− ← build neg rewards in Standard Positive Form to avoid collision and terminal well
            else if final approach phase then
                P+ ← build pos rewards at vertiport
                P− ← build neg rewards in Standard Positive Form to avoid collision
            // Use the FastMDP algorithm to solve for the next action and determine the next state
            s_{t+1} = FastMDP(s_t, A, L, P+, P−)
            S_{t+1}[aircraft] ← s_{t+1}
        // Now that all aircraft have selected an action, apply it
        S_t ← S_{t+1}
```

shown in Figure 4. The simulation covers a 25km by 25km by 25km volume which contains a configurable number of aircraft and gates. The location of the gates are chosen to lie at regular intervals along a circle surrounding the vertiport.

Simulation begins with all aircraft spawned randomly around the terminal area. Each aircraft creates and solves its own MDP at each timestep, and then uses the solution of the MDP to select its next action. The actions of all aircraft are performed simultaneously in the simulation at the beginning of the next time step, simulation then advances by one time step (0.1 seconds), and the vertiport terminal area controller (VTAC) state machine is updated. During the simulation, we track the number of near mid-air collisions (NMACs) that occur.

## IV. Results

Simulations were run for two environments:

1) 2 rings, 15 aircraft
    - vertiport capacity 1
    - ring 1 capacity 2 with radius 2000
    - ring 2 capacity 6 with radius 4000
    - approach threshold radius of 5500
2) 4 rings, 45 aircraft
    - vertiport capacity 1
    - ring 1 capacity 8 with radius 2000
    - ring 2 capacity 12 with radius 3500
    - ring 3 capacity 15 with radius 5000
    - ring 4 capacity 20 with radius 6500

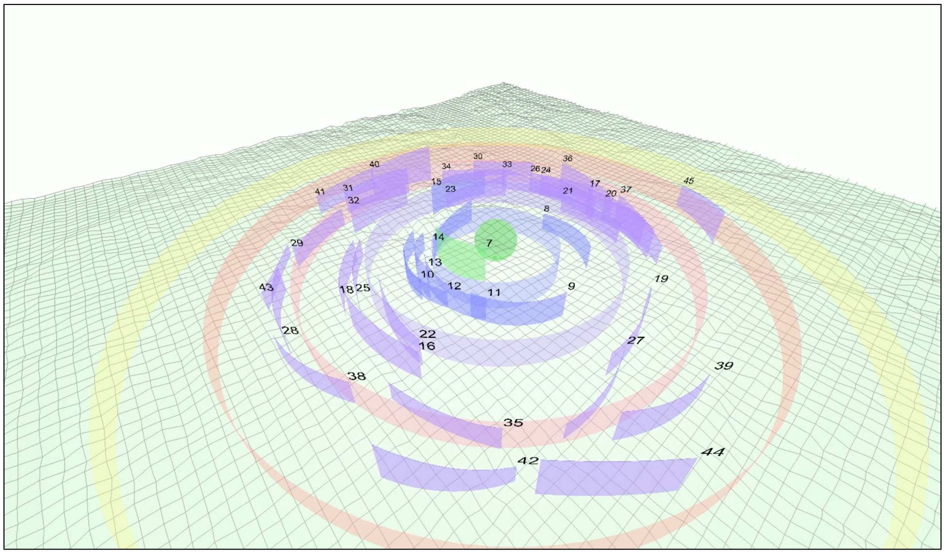


**Fig. 6 Screenshot of simulation showing a vertiport terminal area with four rings and 45 aircraft.**

- approach threshold radius of 7500

The aircraft positions were randomly initialized to lie in uniformly distributed in a region around the vertiport and were initialized with random initial headings. The number of aircraft were chosen to be less than the total capacity of the VTAC capacity $C$.

Simulation results demonstrate that the algorithm is able to successfully sequence aircraft without collisions. An example video with 15 aircraft and 2 rings is available at the URL below which demonstrates the airspace design, state machine, and priority sequencing. The aircraft currently successfully navigate to the arrival gates, are sequenced by the VTAC, and then follow the ring based airspace design while approaching the vertiport. Over 10 runs of this setup with randomized initial conditions 0 NMACs occurred and 0 collisions occurred.

Video: `https://youtu.be/J2CNj5VsHQ4`

Figure 6 shows a screen shot of a simulation with 4 rings and 45 aircraft with a video available at the following URL. This example shows the difficulty of sequencing large numbers of aircraft with increased density. Despite the complexity in a single run 9 NMACs occurred and 0 collisions occurred. The video is rendered with additional indicators showning the intent of each aircraft. Green lines extend from each aircraft to its goal and help understand the decision making that occurs.

Note that the 9 NMACs that occur appear to be caused when aircraft transition from an outer ring to an inner ring. Sometimes this results in what appears to be an undesirable density resulting from excessive closure rate to the ring. Once the closure happens, the rate at which the aircraft separate also appears to be too low. This needs to be investigated in future work to see if a change to the algorithm can better manage this issue.

Video: `https://youtu.be/P8R_JaB5pdI`

## V. Conclusion

In this paper we have demonstrated an airspace design concept to handle an uncertain number of UAM aircraft approaching a vertiport. The airspace design uses a concentric ring pattern along with queues to sequence the aircraft. The computational burden is split between the vertiport which manages aircraft's assignment to rings and to the final queue and the aircraft which manage their own motion around the rings while avoiding collisions. The result is a hybrid centralized / distributed algorithm in which the aircraft self-organize with little direct coordination required of the vertiport's central controller.

In the future, one way to improve the algorithm will be to alter the first-come-first-served nature of the sequencing to a priority based scheme sorted by a metric such as minutes of fuel/charge remaining. This should improve the safety margin by allowing aircraft which may run out of fuel to land earlier. However, this will introduce additional congestion as a high priority aircraft move towards the inner rings through existing traffic. This may be solved with altitude differences between rings or by introducing speed control.

Additionally, future work may allow for an ability to add new rings dynamically. The code actually currently supports this, but it was found that by expanding the number of rings and thus the approach threshold, some aircraft may be directed to reverse course resulting in unnecessary conflicts as aircraft caught within this newly created ring attempt to sort themselves out. It was decided for this paper to leave this case out and explore at a later time.

Also in general separation could be improved if speed control were part of the guidance provided to each aircraft. Currently all aircraft move at the same max speed in the simulation. Improved separation could result if aircraft close to each other also received additional guidance to change speeds to achieve a better distribution around the ring. It is unclear if this functionality would be better as a centralized vertiport function or if an altered reward structure would allow this behavior to emerge.

# References


[1] Gipson, L., "NASA Embraces Urban Air Mobility, Calls for Market Study," `https://www.nasa.gov/aero/nasa-embraces-urban-air-mobility`, 2017. Accessed: 2018-01-19.

[2] "Uber Elevate | The Future of Urban Air Transport," `https://www.uber.com/info/elevate/`, 2017. Accessed: 2018-08-13.

[3] Holden, J., and Goel, N., "Fast-Forwarding to a Future of On-Demand Urban Air Transportation," *San Francisco, CA*, 2016.

[4] "Urban Air Mobility," `http://publicaffairs.airbus.com/default/public-affairs/int/en/our-topics/Urban-Air-Mobility.html`, 2018. Accessed: 2018-08-13.

[5] "Future of Urban Mobility," `http://www.airbus.com/newsroom/news/en/2016/12/My-Kind-Of-Flyover.html`, 2017. Accessed: 2018-08-13.

[6] Kleinbekman, I. C., Mitici, M. A., and Wei, P., "eVTOL Arrival Sequencing and Scheduling for On-Demand Urban Air Mobility," *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, IEEE, 2018, pp. 1–7.

[7] Pradeep, P., and Wei, P., "Energy Optimal Speed Profile for Arrival of Tandem Tilt-Wing eVTOL Aircraft with RTA Constraint," *IEEE CSAA Guidance, Navigation and Control Conference*, 2018.

[8] Bosson, C., and Lauderdale, T. A., "Simulation evaluations of an autonomous urban air mobility network management and separation service," *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3365.

[9] Holden, J., and Goel, N., "Fast-forwarding to a future of on-demand urban air transportation," *San Francisco, CA*, 2016.

[10] Wei, P., Surakitbanharn, C., Landry, S., and Sun, D., "Workload evaluation of sectorized air traffic control and stream management," *2012 Integrated Communications, Navigation and Surveillance Conference*, IEEE, 2012, pp. N2–1.

[11] FAA, "Optimization of Airspace and Procedures in the Metroplex (OAPM) Study Team Final Report, North Texas Metroplex," *Proceedings of the 2011 Global PBN Summit*, 2011.

[12] Bertram, J. R., Yang, X., Brittain, M., and Wei, P., "Online Flight Planner with Dynamic Obstacles for Urban Air Mobility," *2019 Aviation Technology, Integration, and Operations Conference*, 2019.

[13] Bertram, J. R., and Wei, P., "Distributed Computational Guidance for High-Density Urban Air Mobility with Cooperative and Non-Cooperative Collision Avoidance," *AIAA SciTech 60th Aerospace Sciences Meeting*, 2020.

[14] Sutton, R. S., and Barto, A. G., *Reinforcement learning: An introduction*, Vol. 1, MIT press Cambridge, 1998.